TagMaster

MANUAL

# *LR-series User's Manual*

Doc no. 06-118 01
© TagMaster AB

GEN4 LONG-RANGE RFID

**Copyright**
The copyright and ownership of this document belongs to TagMaster AB. The document may be downloaded or copied provided that all copies contain the full information from the complete document. All other copying requires a written approval from TagMaster AB.

**Disclaimer**
While effort has been taken to ensure the accuracy of the information in this document TagMaster AB assume no responsibility for any errors or omissions, or for damages resulting from the use of the information contained herein. The information in this document is subject to change without notice.

**1 Introduction 5**

**2 TagMaster RFID System Overview 6**
2.1 ID-tags ... 7
2.2 Reader ... 7
2.3 Host Computer ... 7
2.4 GEN4 Platform ... 8

**3 Hardware Overview 9**
3.1 Controller Board ... 9
3.2 RF-block ... 10
3.3 Expansion Block ... 11

**4 Software Overview 12**
4.1 Reader System Software ... 12
4.2 Reader Application Software ... 16
4.3 Software Development ... 17

**5 Interfaces 20**
5.1 Service Interface ... 20
5.2 Serial Communication Interfaces ... 20
5.3 Ethernet ... 21
5.4 Wiegand/Mag-stripe ... 21
5.5 USB Host ... 21
5.6 Micro SD memory Card ... 22
5.7 Inputs and Outputs ... 22
5.8 Indicators ... 23
5.9 Expansion Block Interface ... 23

**6 Features 24**
6.1 Web Server ... 24
6.2 Embedded SQL Database ... 24
6.3 Frequency Hopping ... 24
6.4 Real-Time Clock ... 24
6.5 Watchdog Timer ... 24
6.6 Writing ID-tags ... 25
6.7 Motion Detection ... 25

**7 Configuration and Maintenance 26**
7.1 Linux Shell ... 26
7.2 Logging on ... 28
7.3 System Settings ... 30
7.4 Managing Linux Files ... 31
7.5 Managing Linux Processes ... 34
7.6 Configuration Scripts ... 35
7.7 Network Configuration ... 37
7.8 Mounting Storage Devices ... 42
7.9 SQLite Database Management ... 43
7.10 System Information ... 44

**8 Software Updates 47**
8.1 Updating Files in the File System ... 47
8.2 Upgrade Flash Memory using U-Boot ... 48
8.3 Restore U-Boot or Initial Boot Loader ... 50

**9 Contact 51**
9.1 Technical Support ... 51
9.2 Office ... 51

**10 Glossary 52**
**11 References 54**
**12 Online Resources 55**
**Appendix A Useful Software Tools 1**

# 1 Introduction

This section introduces TagMaster's LR-series Reader platform and defines the target group for this manual.

The LR-series Reader is a long range Radio Frequency Identification (RFID) device for remote identification of ID-tags. The Reader can be used as a stand-alone reading device, or incorporated in more comprehensive identification systems through several standard interfaces.

LR-series Reader system software is based on the GNU/Linux operating system, which is a platform suitable for development of custom-made Reader application software.

The aim of the manual is to present a system integrator, who has the necessary education and training, with the information needed to correctly integrate the Reader with an identification system and expand the Reader's conventional area of use.

This manual does not describe any mechanical installation aspects. Refer to the LR-6 Reader Data Sheet [1] and LR-series Installation Manual [2] for more information about physical dimensions, electrical characterization, certifications, and physical installation.

© TagMaster AB 

# 2 TagMaster RFID System Overview

This section describes the elements of a identification system based on TagMaster's products.

Identification systems differ depending on the application, but the two main elements of all TagMaster identification system are the Reader and the ID-tag. Peripheral elements are for example a host computer and other external devices, such as traffic lights and barriers.

The Reader can be connected to a host computer using the standard serial communication interfaces RS232 or RS485. The Reader can also be connected through Ethernet to a TCP/IP network or incorporated in existing access control systems using the Wiegand/Mag-stripe interface.

A host can hold some of the operational logic of the identification system, for instance logging read ID-tags or keeping database entries for every ID-tag that has access to a restricted area. Installed with adequate application software, the Reader is also capable of operating stand-alone. Events registered by the Reader are time stamped using a battery-backed up real-time clock.

The Reader has several input and output options. When an acceptable ID-tag is read, the Reader can pull a relay connected to a barrier and grant access to a vehicle and at the same time, for instance sound the buzzer and flash the visible indicators with green light.

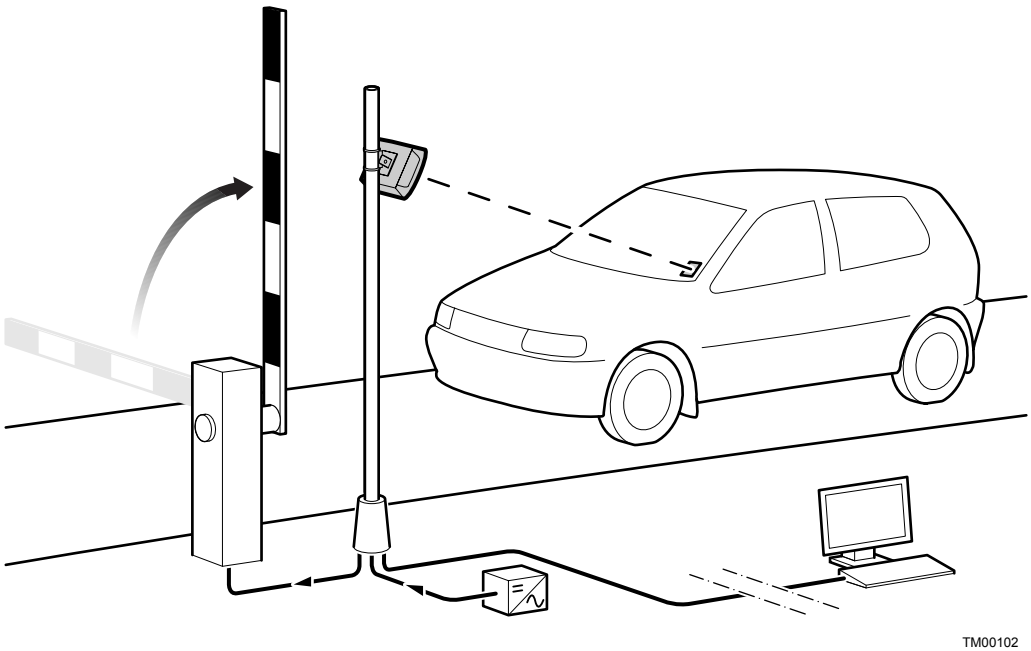


*Figure 1 Identification system overview*

The figure above shows the overview of a TagMaster vehicle access control system with a pole-mounted Reader, an ID-tag mounted on the inside of a car windscreen, a host computer, a power supply, and a barrier.

## 2.1 ID-tags

An ID-tag carries information that can be read at a long distance using radio frequencies. The actual reading range depends on Reader type, the ID-tag, Reader configuration setting, and environmental conditions.

Standard ID-tags have the shape of a credit card, but are slightly thicker. Every ID-tag has a unique and permanent identification number called ID-tag mark.

Many ID-tags can be read concurrently. A lithium cell is used in the ID-tag to preserve stored data, and get a high communication speed.

There are two general types of ID-tags called ScriptTags and MarkTags. ScriptTags can both be read from and written to, while MarkTags only can be read. The data stored in a MarkTag includes the ID-tag mark. The data stored in a ScriptTag includes the ID-tag mark and a writable data field called ID-tag user data.

The data in an ID-tag include a 32-bit checksum for automatic verification. The Reader does not report ID-tag readings containing invalid checksums. Only valid ID-tag data is reported in order to increase the system and application software security.

The LR-series Reader is capable of communicating with all TagMaster ID-tag formats, which include the following formats:

- Read-only
- Read-and-write
- High or low data speed
- Random and constant mode
- ID-tags with mini-, quarter-, or full-memory data sizes
- TagMaster's Open32 ID-tag format

Product sheets for corresponding ID-tags [3], [4], [5], [6], and [7] present key features and technical data.

## 2.2 Reader

The Reader is a device for reading ID-tags using 2.45 GHz radio frequency. In addition to reading, some Readers also have the capability to write information to ScriptTags. The Reader is also capable of detecting moving objects without ID-tags, so called movement detection.

The Reader has built-in antennas for communication with ID-tags as well as various interfaces for communication with a host computer and other external devices.

To reduce the risk of interference, several Readers in close proximity to each other are set to different frequency channels. Readers can use frequency hopping when one specific frequency is not determined.

The LR-series Installation Manual [2] presents an installation engineer with the information needed to correctly install the LR-series Reader.

## 2.3 Host Computer

A host computer is the external intelligence, which can act as master of a Reader or a set of Readers.

Some or all of the Reader functionality can be placed externally in a host. The communication between the Reader and the host computer is done over Ethernet or a serial communication interface.

A temporarily connected PC is used to configure the settings of the Reader. The configuration is performed using either a standard web browser or terminal emulation software, depending on the installed Reader application software.

## 2.4 GEN4 Platform

TagMaster's fourth generation Reader platform is called GEN4. The aim of GEN4 is to incorporate open standards in the design of the Reader, regarding both hardware and software.

The GEN4 platform is based on Linux which facilitates integration and adaptation of the Reader to highly specific requirements. For this purpose TagMaster also supplies a Software Development Kit (SDK). The SDK is a complete set of tool for developing customer-specific applications software, which can be downloaded to the Reader and executed in the Reader's Linux environment.

GEN4 Readers are also provided with two USB host interfaces and SD memory card slot, which make it possible to add additional hardware and storage media.

The TCP/IP network functionality of the Reader enables remote operation and maintenance. The Reader can also operate completely stand-alone using the built-in database that stores the approved ID-tag identities. In this type of configuration the Reader can, for example, receive input signals from a loop detector and control a door or a barrier directly.

GEN4 Readers are backwards compatible with older revision Readers regarding previous channel frequency allocation scheme to facilitate integration with systems containing older revision Readers.

© TagMaster AB 

# 3 Hardware Overview

This section describes the Reader hardware. On a general level, the Reader consists of two physical building blocks, which are the RF-block and the controller board.

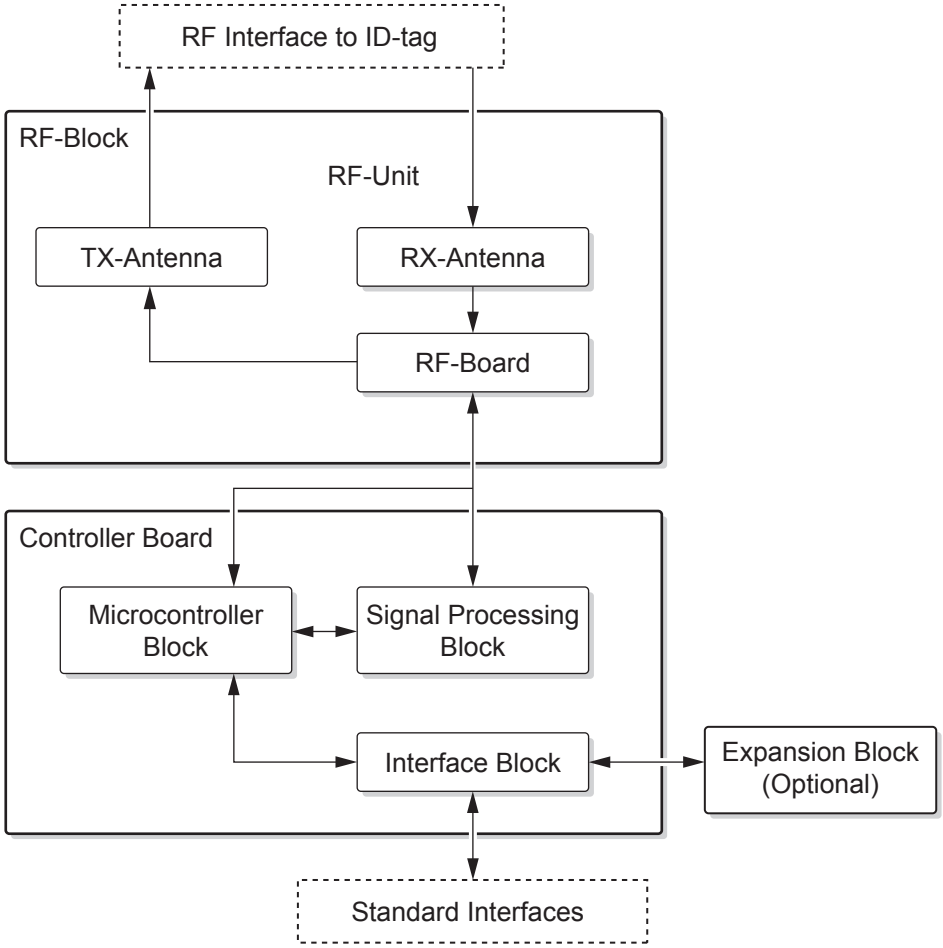


*Figure 2 Hardware architecture overview*

The RF-block and controller board subsequently consist of several other blocks. The building blocks constitute the Reader platform on which different Reader products are based. Each building block is described in the subsections below.

## 3.1 Controller Board

The controller board is the main component of the Reader and it controls the behaviour of the Reader. The controller board is conceptually divided into a microcontroller block, a signal processing block and an interface block.

### 3.1.1 Microcontroller Block

The microcontroller block is built around a Microcontroller Unit (MCU) from Atmel, the AT91RM9200 MCU with an ARM9 CPU core. The microcontroller block has Flash memory, SDRAM memory, and a real-time clock with battery backup.

The MCU runs a standard Linux operating system, including drivers and services. See section 4 Software for more information about the operating system.

*Table 1 Controller board memory*

| Memory | Size | Description |
|---|---|---|
| RAM | 32 MB | Volatile memory |
| Flash memory | 16 MB | Holds all software and is accessed via the Journaling Flash File System version 2 (JFFS2) |

The MCU has an internal watchdog timer that can be set to restart the controller board in case of system malfunction, as described in section 6.5 Watchdog Timer.

For further information about the Reader file system, see section 4.1.4 File System.

### 3.1.2 Signal Processing Block

The signal processing block on the controller board receives radio-signals from the RF-board. The signal processing block performs analogue filtering and analogue to digital conversion. The digitally converted signal is fed to a Field Programmable Gate Array (FPGA), which filters the signal digitally and decodes the ID-tag data.

### 3.1.3 Interface Block

External devices can be connected via several standard communication interfaces to the interface block on the controller board. If the standard interfaces are insufficient, new interfaces and even new functionality can be added via the expansion block.

For further information about expansion possibilities and available interfaces, see section 5 Interfaces.

## 3.2 RF-block

The RF-block is the radio interface and it is primarily used for communication with ID-tags. The RF-block consists of an RF-board, an RX-antenna, and a TX-antenna.

The RF-block is controlled by software (frequency, attenuation, amplification, output power, and so forth), which is sometimes referred to as Software Defined Radio (SDR).

A Reader can be set to a radio frequencies within the operating frequency band. Two Readers set to different frequencies will not interfere, even if installed in close proximity of each other. For instance, a Reader set to 2455.0 MHz does not interfere with a Reader set to 2455.1 MHz.

Reader models prior to GEN4 use a frequency channel allocation scheme that divides the frequency band into distinct channels separated by 300 kHz.

GEN4 based Readers are backwards compatible with the previous channel frequency allocation scheme to facilitate integration with installations of older revision Readers.

Frequency hopping is also available with the LR-series Reader, see section 6.3 Frequency Hopping.

## 3.3 Expansion Block

The expansion block is designed to meet requirements not met by the standard hardware platform. The expansion block can add specific hardware capabilities and interfaces to a standard Reader.

The expansion block is connected to the controller board via the expansion block interface. The Reader expansion block interface consists of several standard interfaces and is described in section 5.9 Expansion Block Interface.

# 4 Software Overview

This section describes the Reader software. On a general level, the Reader has two software layers, which are system software and application software.

The Reader system software constitutes the foundation of the Reader from a software perspective. All GEN4 Readers have system software installed by default.

Different Reader application software resides on top of the system software. The Reader application software defines the distinguished behaviour of the Reader.

## 4.1 Reader System Software

The system software consists of the following components:

- Boot-loaders
- GNU/Linux operating system
- File system
- Drivers
- Utilities
- Software libraries
- Configuration files and scripts

### 4.1.1 Boot Loaders

In general terms, a boot loader is a small piece of software that executes soon after powering up a Reader.

In a Reader, there are actually two boot loaders. The first boot loader is loaded and performs the initial system configuration, then de-compress and loads the main boot loader, which is U-Boot.

U-Boot is an universal boot loader, see <http://u-boot.sourceforge.net>. It supports a majority of embedded architectures and is released as open source under the GPL license. To reduce the memory foot-print, U-Boot is stored as a gzip-compressed binary referred to as the U-Boot image.

U-Boot is responsible for unpacking the Linux kernel and pass parameters to the Linux kernel before it is started.

In addition to de-compressing and starting Linux, U-Boot also presents a simple terminal-based user interface that is accessed via the service interface. During the Reader boot process, U-Boot will output boot information. The boot process can be manually interrupted, giving access to the terminal user interface.

Via the U-Boot user interface it is possible to perform complete system upgrades and do low-level configuration, as described in section 8.2 Upgrade Flash Memory using U-Boot.

Note however that in most cases there is no need for accessing the U-Boot user interface. Changing the U-Boot settings is not recommended.

### 4.1.2 Operating System

The Linux kernel is the innermost part of the GNU/Linux operating system on which the Reader system software is based. The kernel is the layer that resides between hardware and user space where application software executes. When started, the kernel mounts the file system and starts the init process as described in section 7.6.1 Init Script.

The operating system is based on the official Linux kernel, see <http://www.kernel.org>. The kernel is patched with the official AT91RM9200 patch appropriate for the specific microcontroller used by the microcontroller block.

The kernel version installed on a Reader is outputted just after the kernel image has been de-compressed by U-boot. The kernel version is also made known by using the command "uname", as described in section 7.10.2 Linux Kernel Version. Updating the kernel is possible using U-Boot. The current kernel version is likely to change with new system software releases.

The Linux kernel has been kept as intact as possible. Only minor modifications that are required by the GEN4 hardware, such as definitions of serial interfaces ttySx numbers, have been introduced.

### 4.1.3 Memory Protection

The operating system has user space memory protection, preventing one process from corrupting the memory of another process running simultaneously on the same Reader. It provides a stable and secure software platform, on which an integrator can build and develop custom software components.

The MCU has Memory Management Unit (MMU), responsible for handling memory access requests. The Reader has no swap memory, because Flash wears out quickly and is slow.

### 4.1.4 File System

When started, the Linux kernel mounts a root file system, which contains a directory structure and files. The root file system contains binaries (application software, commands, libraries, and so forth) and configuration files that are required by the Reader system.

The root file system hierarchy complies with common practice as showed in the table below. There is one major exception, the directory "/tag/", which contains TagMaster specific files.

*Table 2 Reader root file system hierarchy*

| Directory | Description |
|---|---|
| / | The root |
| /bin/ | Essential user command binaries |
| /dev/ | Device files |
| /etc/ | System configuration files |
| /lib/ | Essential shared libraries |
| /mnt/ | Mount points for temporarily mounted file systems |
| /opt/ | Add-on software packages |
| /proc/ | Virtual file system in RAM for kernel and process information |
| /root/ | Root user's home directory |
| /sbin/ | System binaries |
| /tag/ | Secondary hierarchy containing TagMaster specific files |
| /tag/bin/ | TagMaster binaries |
| /tag/etc/ | TagMaster configuration and start-up files |
| /tag/lib/ | TagMaster libraries |
| /tag/var/ | TagMaster variable data, such as database files for standard Reader software applications |
| /tmp/ | Temporary files |
| /usr/ | Secondary hierarchy containing binaries and documents useful to most users |
| /usr/bin/ | Most user commands |
| /usr/lib/ | User libraries |
| /usr/sbin/ | Non-vital system binaries |

See section 7.4 Managing Linux Files for information on how to navigate in the file system and how to rename, copy, and edit files.

For non-volatile storage, application software store data onto a Flash based device via the file system, which is optimised for Flash usage.

### 4.1.5 TagMaster System Software

taglib, TAGP, tagd and tagmod are components of the system software that are specific to TagMaster.

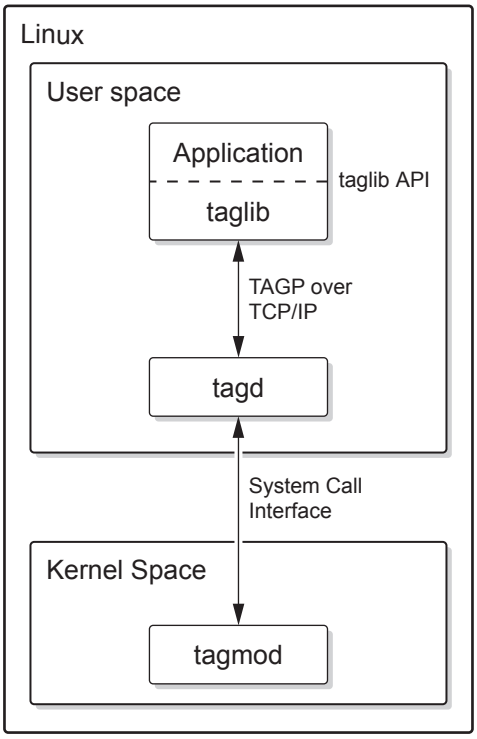


*Figure 3 TagMaster system software overview*

taglib is a software library, which presents an application programming interface to TagMaster's GEN4 Readers. taglib provides easy to use function calls that perform tasks such as reading information from ID-tags, turning the indicators on or off, and configuring Reader settings.

taglib provides a framework to manage several Readers simultaneously from one application process. taglib implements a communication protocol called TAGP. taglib is flexible and can be used together with other existing libraries, for example Linux APIs used for accessing serial interfaces, USB interfaces, and so forth.

Readers have an application abstraction layer that software designers utilise for application development. The abstraction layer is a high-level daemon process called tagd. The daemon process is used to control Reader properties such as frequency channel, buzzer, and relay state.

From an application designer's point of view, tagd is a server handling the Reader's resources. Using the same perspective, the application is a client connected to the tagd server requesting access to the Reader's resources. Several clients, or peers, can be connected to the same tagd server. tagd communicates using TAGP. Only one tagd can run on each physical Reader.

tagd supports multiple connections and several clients can therefore be connect simultaneously. For multi-client solutions, the tagd daemon supports a lock feature that

allows a specific client to take full control over the Reader, leaving other clients with only limited access.

The kernel module tagmod is the TagMaster specific hardware interface controlling the RF-block, the reading and writing of ID-tags, and so forth. Writing Reader application software that communicates directly with tagmod is not possible. Instead, tagmod and Reader hardware is controlled indirectly by application software through tagd.

Application software, which for instance reads ID-tags or sets a Reader channel, connects to tagd via communication sockets. After handshaking, application software receives events as soon as the tagd daemon gets them from the tagmod kernel module.

## 4.2 Reader Application Software

Available application software for the GEN4 Reader are PassMan, WatchMan and WiseMan. The Reader application software determines how the Reader can be configured, how it operates and how it communicates with the system to which it is connected.

See the table below for a software overview and comparison between different Reader application software. Note that software features might be subject to change or available only as options.

*Table 3 Software comparison*

| Feature | WiseMan | WatchMan | PassMan |
|---|---|---|---|
| User-configurable for maximum flexibility | ☑ | ☑ | ☑ |
| Movement detection | ☑ | ☑ | ☑ |
| User-selectable radio frequency channels | ☑ | ☑ | ☑ |
| Frequency hopping | ☑ | ☑ | ☑ |
| Auto-channel detection | ☑ | ☑ | ☑ |
| Controllable from host computer | ☑ | ☑ | ☑ |
| Serial interface (RS232 and RS485) implementation | ☑ | ☑ | ☑ |
| Wiegand and Mag-stripe interface implementation | | | ☑ |
| ID-tag writing capabilities (applies only to the LR-6 Reader) | ☑ | ☑ | |
| Web based configuration interface | ☑ | ☑ | |
| Stand-alone capabilities | ☑ | | |
| Utilises internal database | ☑ | | |
| Logging | ☑ | | |

Reader application software delivered by TagMaster are copy-protected, meaning that the application software can only be executed from the Reader on which it is installed. Moving a piece of pre-installed application software from one Reader to a second Reader and then executing it on the second Reader is not possible.

### 4.2.1 WiseMan

The WiseMan software is used in identification systems that require a Reader with both information storage and decision-making capabilities.

A Reader installed with WiseMan can operate stand-alone. As an alternative to using the Reader stand-alone, is to control the Reader from a host computer and place some or all of the functionality externally in the host. The communication between the WiseMan software and the host application software is done over a serial communication interface.

For further information about the WiseMan application software, see WiseMan Software Manual [8].

### 4.2.2 WatchMan

The WatchMan software is used in identification systems that include one or several Readers controlled from a host computer.

The WatchMan software is suited for identification systems where the Reader is controlled from a host computer that handles functionality customised by the customer. The communication between the WatchMan software and the host application software is done over a serial communication interface.

For further information about the WatchMan application software, see WatchMan Software Manuel [9].

### 4.2.3 PassMan

The PassMan software makes it possible to incorporate the Reader in an existing access control system using Wiegand or Mag-stripe communication interfaces. The PassMan software also makes it possible to control the Reader from a host computer using the serial communication interface.

The Reader installed with PassMan has only limited capabilities to make decisions on its own. The Reader application software cannot store any information about read ID-tags. Functionality is foremost placed in the access control system or the host computer.

For further information about the PassMan application software, see PassMan Software Manual [10].

## 4.3 Software Development

The GEN4 platform facilitates custom software development. Developing Reader application software assumes that the person developing software is fluent in C-programming and has experience with the Linux environment.

Developing custom application software is of interest when standard applications software, as described in section 4.2 Reader Application Software, is insufficient and the Reader is implemented in existing system with specific needs.

### 4.3.1 Checklist

When developing Reader application software, several aspects about the system in which the Reader will be incorporated must be taken into requirements.

As a guide when planning the development of application software for the GEN4 Reader, consider the following checklist:

- Should the application software reside and run in a stand-alone Reader or should the Reader be connected to a host?
- Should the Reader be controlled by a host using the TAGP protocol? Consult the TAGP Protocol Specification [11].
- Should the Reader save data in the database?
- Should the application software be made highly versatile using the taglib software library? Consult the taglib Software Library Specification [12].

If the following conditions are met, developing new application software is not necessarily required:

- The Reader runs any of the standard application software which communicates using ConfiTalk? Consult the ConfiTalk Reference Manual [13].
- Configuring existing application software is an adequate solution? Consult the corresponding software manual [8], [9], and [10].

### 4.3.2 Software Development Kit

A Software Development Kit (SDK) for GEN4 based Readers is available from TagMaster. The SDK is based on open source components and software maintained by TagMaster.

The SDK is a complete package for developing software for both Readers and hosts running Linux, containing the following components:

- Cross-compilers for Linux
- Libraries for Linux
- Debuggers
- Source code examples
- Documentation

Developing software for Windows hosts using ConfiLib is possible and applies for Readers running WiseMan or WatchMan. But it is not possible to reuse existing binaries from Readers with versions older than GEN4.

The SDK utilise tools and libraries from the GNU Compiler Collection (GCC) project, which is an open source project. GCC is one of the most widely used compiler collections; it is available for most platforms.

The preferred programming language is C for several reasons; GNU/Linux is mostly written in C, most software for Linux is also written in C, and C provides full control.

The preferred development platform is Linux. Test and debug a newly developed application on host before downloading it to the Reader.

For information about how to install the SDK, see Software Development Kit Manual [14].

### 4.3.3 Adding New Drivers

Adding software support for new hardware, such as USB devices, is possible by building new kernel modules. Kernel modules are object files that contain code to extend the running kernel. Including a device driver in the kernel itself is not possible because it requires that the whole kernel is rebuilt.

When selecting a new USB device that the Reader should use, such as a WLAN or UMTS modem, the recommendation is to modify an existing device driver that is already supported by Linux. Building new kernel modules is quite difficult and should only be undertaken by experienced Linux users. Developing a driver from scratch is complex and difficult.

Most likely new drivers must be re-compiled for the Reader. The source code for the new driver and the kernel that the driver is built for must be available in order to build the driver.

Building new kernel modules is possible by using the cross-compiler provided with the SDK.

# 5 Interfaces

This section describes the standard interfaces available with the Reader. Interfaces include communication interfaces, inputs and outputs, indicators and so forth. Each interface and how the interface is accessed using the Reader software is briefly described.

For electrical specifications regarding the interfaces and pin-numbering, see the LR-series Reader Installation Manual [2].

## 5.1 Service Interface

The controller board has an RS232 serial interface that serves as a service interface. As the interface name suggests, the service interface is used for service and maintenance. Logging on to the Linux system and having full access to the Reader is possible through the service interface. Updating software via the service interface is also possible.

The device-handle for the service interface is "/dev/ttyS0".

The service interface communication settings, expressed in the table below, are static in order to facilitate costumer support.

*Table 4 Service interface communication settings*

| **Setting** | **Static Value** |
|---|---|
| Baud rate | 115200 baud |
| Data bits | 8 |
| Parity | None |
| Stop bits | 1 |
| Flow control | None |

**Note**: Do not make the service interface accessible from outside the Reader enclosure and do not use the service interface as an application port, for instance to implement a custom serial protocol. Doing so will compromise system security.

## 5.2 Serial Communication Interfaces

The Reader has both an RS485 serial communication interface and an RS232 serial communication interface. If additional serial communication interfaces are required, the expansion block interface can be used (see section 5.9 Expansion Block Interface).

Both interfaces can be individually configured regarding baud rate, data bits, stop bits, parity bits, and so forth. See the corresponding product data sheet for more information about the supported settings.

These interfaces can be used by Reader application software that implements custom serial protocols to interact with other devices or with a host.

### 5.2.1 RS232 Serial Interface

The Reader has an RS232 serial interface, which is a single-ended full-duplex interface.

The RS232 interface is supported by drivers statically built into the Linux kernel. It is accessed via standard Linux system calls using the device-handle "/dev/ttyS1".

### 5.2.2 RS485 Serial Interface

RS485 supports multi-drop serial networks. The communication can be in either full duplex of half duplex, that is 4-wire or 2-wire communication. Signalling is differential and the RS485 interface is therefore more robust and less sensitive to transmission errors compared to a common RS232 interface.

The RS485 interface is supported by drivers statically built into the Linux kernel. It is accessed via standard Linux system calls using the device-handle "/dev/ttyS2".

The duplex setting is controlled via software, no hardware straps exist. The duplex setting, full or half duplex, is set via TAGP command messages or taglib function calls.

## 5.3 Ethernet

The Reader has a 10/100 Mbps Fast Ethernet interface. The network interface supports auto-negotiation for automatic media speed and protocol selection. Indicators on the Ethernet connector show link activity and link speed.

The Ethernet interface has full support in the Linux kernel. The interface is configurable with the command line tool ifconfig; see section 7.7 Network Configuration for more information.

By default a Reader has several services that use the Ethernet interface, such as HTTP, SSH and tagd. Custom Reader application software can apply standard Linux functions to use the network interface.

## 5.4 Wiegand/Mag-stripe

The Reader has an access control interface that supports both Wiegand and Mag-stripe signalling. The Wiegand/Mag-stripe interface is uni-directional and consists of three outputs.

Because Wiegand and Mag-stripe share the same physical interface only one can be used at any given time. It is also possible to use the three output signals individually if more output pins are required.

Data can be sent to the Wiegand/Mag-stripe interface through TAGP messages or taglib function calls. The software API provides a raw access to the interface and consequently parity, start-patterns and other protocol-specific fields must be handled by the application using the interface.

## 5.5 USB Host

The controller board has two USB host interfaces that comply with the USB 2.0 Full Speed standard. The USB host interfaces can be used as interfaces for expanding the functionality of a Reader when the standard functionality is insufficient, adding for instance Flash memory, a GSM modem or a WLAN interface.

The Reader is pre-installed with Linux drivers that can be used to access USB memory storage devices. See section 7.8.1 Mounting USB Devices for more information on how to use and access an USB memory storage device.

Additional device drivers must be developed in order to use the USB host interface with other devices. See section 4.3.3 Adding New Drivers for information about adding device drivers.

## 5.6 Micro SD memory Card

The controller board has a Secure Digital (SD) memory card interface, which supports physical form factor of micro SD, also known as TransFlash. The interface can be used to add more Flash memory to a Reader.

The Reader is pre-installed with Linux drivers that can be used to access memory cards. See section 7.8.2 Mounting Micro SD Devices for more information on how to use and access an SD memory card.

Custom Reader software applications could for instance be installed and executed from a SD memory card. Upgrading the software is then just a matter of changing memory card.

## 5.7 Inputs and Outputs

The Reader has several inputs and outputs, which can be used to incorporate the Reader in a system.

*Table 5 Inputs and outputs*

| Interface | Description |
|---|---|
| Inputs | The Reader has three optocoupler inputs. The inputs can be set to generate a software event on change. |
| Outputs | The Reader has two open collector outputs that can be set high or low.<br><br>The three Wiegand signals are also possible to control with software and use as output |
| Relay output | The Reader has one relay output for heavy duty loads, which can be set open or close. |
| Tampering switches | The Reader has two mechanical tampering switches that break if the cover of the Reader is opened. The first tamper switch is connected internally to the controller board and creates a software event when opened and closed.<br><br>The second tamper switch has an interface which can be connected to an external alarm loop. |

All inputs and outputs, except the external tampering switch, are supported by TAGP and taglib.

If additional inputs and outputs are required, it is possible to use the expansion block interface (see section 5.9 Expansion Block Interface).

## 5.8 Indicators

The Reader has visual indicators that can show red or green, or both, which is referred to as yellow. The indicators can also be turned off. The Reader also has a buzzer for audible indication.

Visual indicators and the buzzer are supported by TAGP protocol messages and taglib function calls.

## 5.9 Expansion Block Interface

The expansion block interface can be used to add an expansion block to a Reader. All expansion block interfaces have built-in software support. An expansion block in its simplest form could just be an additional serial interface, which would consist of ESD protection, serial interface drivers, and an interface socket.

*Table 6 Expansion interfaces*

| Interface | Description |
|---|---|
| TWI | Two-Wire Interface (TWI) is an interface with one data and one clock signal. TWI is essentially the same interface as Inter-Integrated Circuit (I2C). All devices participating in the bus are connected together and each device is addressed by a unique identifier.<br><br>This interface is supported by drivers statically built into the Linux kernel. To fully take advantage of the TWI interface it is necessary to develop a device specific driver. |
| SPI | The Serial Peripheral Interface (SPI) is a standard serial bus interface used for communication with peripheral devices. SPI devices communicate using a master-slave relationship, where the Reader is considered master. SPI specifies four signals: Clock, data in, data out and slave select.<br><br>This interface is supported by drivers statically built into the Linux kernel. To fully take advantage of the SPI interface, it is necessary to develop a device specific driver. |
| Two UART | Two Universal Asynchronous Receiver/Transmitter (UART) interfaces that support full-duplex serial communication are available. Each interface has two signals, one for transmission and one for reception.<br><br>These interfaces are supported by drivers statically built into the Linux kernel. They are accessed via standard Linux system calls using the device handles "/dev/ttyS3" and "/dev/ttyS4" respectively. |
| Digital inputs and outputs | Two digital outputs, two digital inputs, and a dedicated input interrupt signal. The inputs and outputs are referenced to ground. The inputs and outputs are supported by TAGP protocol messages and taglib function calls. |
| Power supply | The expansion interface can power an expansion block. Expansion blocks with excessive power consumption are required to have a separate power supply. |

# 6 Features

This section describes some features found in GEN4 Readers.

## 6.1 Web Server

The built-in web server presents a web interface to the Reader. A standard web browser is used to log on to the Reader via Internet or a local network. Status information can be received and configuration settings can be altered when logged on through a web browser.

For information about logging on the Reader using a standard web browser, see section 7.10.1 Web interface.

The web server is a HTTP daemon, which is a very small web server adequate for low-traffic hosts. It implements basic features of standard web servers, including CGI.

## 6.2 Embedded SQL Database

SQLite is a small C library that implements a self-contained, embeddable, zero-configuration SQL database engine.

The SQLite distribution comes with a standalone command line access program (sqlite3) that can be used to administer an SQLite database and which serves as an example of how to use the SQLite library.

## 6.3 Frequency Hopping

The Frequency hopping has a pseudo-random behaviour and is suitable for simple installations where frequency channel planning is not required or desired. Readers using frequency hopping are unaffected by other Readers and unlikely interfered by other surrounding equipment. Sub-bands in which the frequency hopping takes place can be defined using TAGP messages or taglib function calls.

At least 20 Readers can operate in close proximity of each other when all are set to frequency hopping.

When using frequency hopping, the reading performance is slightly reduced.

## 6.4 Real-Time Clock

The Reader has a battery-backed up real-time clock that can present a time stamp with century, year, month, day, day of week, hour, minute, and second. Using the Linux system time, the resolution is 1/1000 of a second.

The battery is charged as long as the Reader unit is powered. The clock will run on battery if the power supply fails.

## 6.5 Watchdog Timer

The microcontroller of the Reader has an internal watchdog timer, which is a hardware timing device that triggers a Reader reset if the system software malfunctions. The intention is to bring the Reader back from a hung state into normal operation.

The system software has to regularly write to the watchdog device (also referred to as "petting the dog").

The watchdog timer is inactive by default. The watchdog is initiated with the command "watchdog". Once enabled, the watchdog command enters a loop and periodically writes to the device "/dev/watchdog". The loop is infinite and cannot be terminated without restarting the Reader.

The watchdog in the microcontroller of the Reader has to be petted every five seconds. Five seconds is also the default value for the watchdog command in the GEN4 implementation.

The Software Development Kit provided by TagMaster has a source code example that covers the use of the watchdog timer for Reader application software.

## 6.6 Writing ID-tags

When writing ID-tags, the Reader occupies a broader spectrum of the frequency band compared to when reading ID-tags. Only one ID-tag at a time is allowed in the reading lobe while writing and frequency hopping must be disabled.

Configure the Reader that writes ID-tags so that it is separated by at least two frequency channels from other Readers in close proximity.

The ID-tag to be written must be still and positioned with the front side directed towards the Reader.

For further information regarding the process of writing ID-tags, see S1251 ScriptTag Data Sheet [3] and S1450 ScriptTag HD Data Sheet [5].

## 6.7 Motion Detection

The built-in motion detector employs the antennas used for ID-tag communication. The Reader makes use of the Doppler effect to detect moving objects. Objects moving in front of the Reader will give rise to a low frequency signal. Approximation of objects speed and direction are derived from calculations based on the signal.

The sensitivity of the motion detection is adjustable.

© TagMaster AB 

# 7 Configuration and Maintenance

This section describes the basic configuration and maintenance of the Reader. Common Linux commands relevant for Reader configuration and maintenance are also described.

The service interface or the Ethernet interface is used for accessing the operating system of the Reader and performing configuration and maintenance.

When logging on for the first time, change the system password and adjust the date and time settings.

Commands described in this manual presuppose that a user is logged on to the Reader operating system using a PC installed with appropriate client applications, such as a terminal emulator, a SSH-client, and a SFTP-client. For suggestions on client applications, see 12Appendix A.

## 7.1 Linux Shell

The Reader can be fully accessed via the service interface using a temporarily connected PC with terminal emulator software. After logging on, the user is presented with a standard Linux shell.

A Linux shell, also called the command line, provides the traditional user interface for the GNU/Linux operating system.

It is also possible to connect to the Reader via Internet or a local network using Secure Shell (SSH). Most aspects of the Reader are controllable using the Linux shell.

A command line user interface can be seen as less intuitive and harder to use compared to a graphical user interface. However, there are a few tricks that make working with the Linux shell easier, for example word completion and command history.

### 7.1.1 Word Completion

The tabulator (tab) key can be pressed for word-completion. When for instance navigating in the file system, there is no need to write the complete file name.

Write the beginning of a file or directory name and press the tab key. If there is more than one file beginning with the same name, these are shown if the tab key is pressed twice.

### 7.1.2 Command History

Arrow key up (↑) and arrow key down (↓) can be pressed to bring up previously entered commands. Using command history, previous commands do not have to be retyped.

### 7.1.3 BusyBox

A utility programs kit called BusyBox is installed on the Reader, providing many common Linux utilities and commands. BusyBox is a single small executable, suitable for embedded systems.

The utilities in BusyBox generally have fewer options than full-featured GNU/Linux commands; however, the options that are included provide the expected functionality and behave very much like their GNU/Linux counterparts.

A full listing of the commands or functions is presented if "busybox" with no arguments is typed at the command line, as follows:

```
tag:~$ busybox
BusyBox v1.2.1 (2006.09.28-11:48+0000) multi-call binary

Usage: busybox [function] [arguments]...
   or: [function] [arguments]...

        BusyBox is a multi-call binary that combines many
        common Unix utilities into a single executable.
        Most people will create a link to busybox for each
        function they wish to use and BusyBox will act
        like whatever it was invoked as!

Currently defined functions:
        [, [[, ash, awk, basename, busybox, cat, chmod,
        chroot, clear, cmp, cp, cut, date, df, dirname,
        dmesg, du, echo, egrep, env, expr, false, fgrep,
        free, getty, grep, halt, hostname, httpd, hwclock,
        ifconfig, ifdown, ifup, init, insmod, kill,
        killall, less, linuxrc, ln, login, ls, lsmod,
        mkdir, mknod, mount, mv, nc, netstat, nohup,
        nslookup, passwd, ping, poweroff, printenv,
        printf, ps, pwd, rdate, reboot, rm, rmmod, route,
        run-parts, rx, sed, sh, sleep, sync, test, top,
        touch, tr, traceroute, true, udhcpc, umount,
        uname, uptime, vi, watchdog, wget, who, whoami
```

Several commands are not covered in this manual. Help for each command is available by typing the command followed by "--help", for instance "date --help", as follows:

```
tag:~$ date --help
BusyBox v1.2.1 (2006.09.28-11:48+0000) multi-call binary

Usage: date [OPTION]... [MMDDhhmm[[CC]YY][.ss]] [+FORMAT]

Displays the current time in the given FORMAT, or sets the
system date.

Options:
        -R              Outputs RFC-822 compliant date
                        string
        -d STRING       Displays time described by STRING,
                        not `now'
        -I[TIMESPEC]    Outputs an ISO-8601 compliant
                        date/time string TIMESPEC=`date'
                        (or missing) for date only,
                        `hours', `minutes', or `seconds'
                        for date and, time to the
                        indicated precision
        -D hint         Use 'hint' as date format, via
                        strptime()
        -s              Sets time described by STRING
        -r FILE         Displays the last modification
                        time of FILE
        -u              Prints or sets Coordinated
                        Universal Time
```

For further information about BusyBox, see <http://www.busybox.net>.

## 7.2 Logging on

The Reader operating system can be accessed, either using a terminal emulator and the service interface (see section 7.2.1) or using an SSH-client and Ethernet (see section 7.2.2). Choose the most convenient communication interface and method when connecting to the Reader.

The same text based user interface to the Reader's shell is presented independent of the two communication interfaces.

### 7.2.1 Logging on Using Service Interface

Log on to the Reader via the service interface using a temporarily connected PC. Use terminal emulation software on the PC, which communicates over the serial interface with the Reader.

Log-on to the Reader as follows:

1. Connect the service interface of the Reader to the serial port of the PC with a RS232 cable.
2. Configure the terminal emulator with the following communication settings (which are the static communication settings for the service interface):
    - 115200 baud
    - 8 data bits

- No parity
- 1 stop bit
- No flow control

3. Open a connection from the text terminal to the Reader. A log-on prompt is displayed. Type "root" as the user name followed by pressing enter. Logging on for the first time, type the default password "qwerty" followed by pressing enter (the password will actually not be displayed when typed at the command line).

```
Welcome to TagMaster Linux
tag login: root
Password: qwerty

login[753]: root login  on `ttyS0'

BusyBox v1.2.1 (2006.09.28-11:48+0000) Built-in shell
(ash)
Enter 'help' for a list of built-in commands.

tag:~$
```

4. After log-on, the command line is displayed and full access to the Reader's operating system is granted.

### 7.2.2 Logging on Using Ethernet

Log on to the Reader via the Ethernet interface using a PC connected to the same network as the Reader. Use a SSH-client to connect to the Reader.

Note that IP address of the Reader must be on a network that is reachable from the PC.

Log-on to the Reader as follows:

1. Configure the SSH-client with the following communication settings:
   - IP address of the Reader (default IP address is 192.168.0.2)
   - Port 22
   - SSH as connection type
2. Open a connection from the SSH-client to the Reader. A command prompt is displayed. Type "root" as the user name followed by pressing enter. Logging on for the first time, type the default password "qwerty" followed by pressing enter (the password will actually not be displayed when typed at the command promt).

```
login as: root
root@192.168.0.2's password: qwerty

BusyBox v1.2.1 (2006.09.28-11:48+0000) Built-in shell
(ash)
Enter 'help' for a list of built-in commands.

tag:~$
```

3. After log-on, the command line is displayed and full access to the Reader operating system is granted.

### 7.2.3 Logging off

Log-off from the operating system before disconnecting the PC from the Reader. To exit the system, type the command "exit" at the command line followed by pressing enter.

```
tag:~$ exit
```

This applies to both service interface and Ethernet interface shells.

## 7.3 System Settings

The following subsections describe how to configure system settings, such as changing system password and adjusting date and time.

### 7.3.1 Setting System Password

Set the system password when logging on to the Reader for the first time.

Keep a record of the password in a secure place. Once changed, the new password will replace the default password or any other previously changed passwords.

Set the system password as follows:

1. Log on to the Reader as described in section 7.2 Logging on. The command line is displayed.
2. Type "passwd" followed by pressing enter. When prompted, type a new password and finally re-enter the new password.

```
tag:~$ passwd
Changing password for root
Enter the new password (minimum of 5, maximum of 8
characters)
Please use a combination of upper and lower case
letters and numbers.
Enter new password:
Re-enter new password:
passwd[749]: password for `root' changed by user
`root'
Password changed.
```

**Note:** The password is saved as an encrypted string in the file system and cannot be obtained from the Reader. The Reader operating system must be reinstalled if the password is lost.

### 7.3.2 Setting Date and Time

The operating system of the Reader has a system clock. The Reader is also provided with a battery-backed up real-time clock. The system clock is set, based on the real-time clock, every time the Reader is restarted.

The process of setting date and time to the real-time clock includes first setting the system clock using the date command or rdate command and then assigning the system date and time to the real-time clock using the hwclock command.

The system clock is set either manually by typing the correct date and time as described in step 2 below, or automatically using a remote time server as described in step 3 below.

Set the Reader date and time as follows:

1. Log-on to the Reader as described in section 7.2 Logging on. The command line will be displayed.
2. Set the system clock manually by typing "date" followed by the current:
   - Month expressed with two digits (00–12)
   - Day of month expressed with two digits (00–31)
   - Hour expressed with to digits (00–23)
   - Minutes expressed with two digits (00–59)
   - Year expressed with four digits.

   The following command would set the system time to 2006-11-28 14:58.

   ```
   tag:~$ date 112814582006
   ```

3. As an alternative to setting the system clock manually, it can be set using a remote time server (presupposed that the Reader has a connection to a network with a RFC 868 time server). Set the system clock automatically using a remote time server by typing "rdate" followed by the host name or IP-address of a time server (Note that the IP address in the example below does not point to a valid time server).

   ```
   tag:~$ rdate 193.15.235.99
   ```

4. Assign the system date and time to the real-time clock by typing "hwclock -w".

   ```
   tag:~$ hwclock -w
   ```

Type "date --help", "rdate --help", or "hwclock --help" for further information about each command.

# 7.4 Managing Linux Files

The following subsections briefly describe common Linux commands used to manage files in the Reader file system.

## 7.4.1 Change Working Directory

The command "cd" is used to change the current working directory.

Type "cd" followed by the relative or absolute path of a directory to change current working directory to the specified directory, as follows:

```
tag:~$ cd /tag/bin/
```

Type "cd " followed by two full stop characters to change current working directory to the parent directory, as follows:

```
tag:~$ cd ..
```

Type "cd " followed by a minus sing to change current working directory to the previous working directory, as follows:

```
tag:~$ cd -
```

### 7.4.2 List

The command "ls" is used to list contents of directories. The command "ls" can be appended with "-l" to get a long and verbose directory listing. Type "ls -l" to get a directory listing, as follows:

```
tag:/tag/bin$ ls –l
-rwxr--r--   1 root  root   129256 Nov 28 23:07 ep2c8.rbf
-rwxr--r--   1 root  root    77478 Dec  1 10:28 passman
-rwxr--r--   1 root  root    94744 Nov 29 15:50 tagd
-rw-r--r--   1 root  root    61050 Nov 29 16:00 tagload.ko
-rw-r--r--   1 root  root   346315 Nov 29 16:00 tagmod.ko
-rwxr--r--   1 root  root    11820 Oct 25 15:56 tmload
-rwxr-----   1 root  root   105705 Nov 29 16:01 watchman
-rwxr-----   1 root  root   118711 Nov 29 16:01 wiseman
```

The long and verbose directory listing contain information about each file, including access permissions, owner, size, modification date, and name.

### 7.4.3 Remove

The command "rm" is used to remove files. Type "rm" followed by the name of the file to be removed, as follows:

```
tag:~$ rm temp_file
```

Use this command with caution. Note that a removed file cannot be restored.

### 7.4.4 Move or Rename

The command "mv" is used to move files from one directory to another. Type "mv" followed by the file name and the destination directory to move a file, as follows:

```
tag:~$ mv temp_file /tmp/
```

The command "mv" is also used to rename files. Type "mv" followed by the old file name and the new file name, as follows:

```
tag:/tmp$ mv temp_file new_file
```

If manipulating files not residing in the working directory, the reference to the file must include the complete path to the file, as follows:

```
tag:~$ mv /tmp/temp_file /tmp/new_file
```

### 7.4.5 Copy

The command "cp" is used to copy files. Type "cp" followed by the file name and the name of the copy, as follows:

```
tag:~$ cp /tmp/new_file executable_file
```

If manipulating files not residing in the working directory, the reference to the file must include the complete path.

### 7.4.6 Change Mode

The command "chmod" is used change the access permissions of files. Permissions include granting the ability to read, write, and execute files. Type "chmod +x" followed by the file name to make a file executable in addition to previous permissions, as follows:

```
tag:~$ chmod +x executable_file
```

Type "chmod --help" for more information about changing permissions of files.

### 7.4.7 Editing Text Files

The operating system of the Reader includes a simple text editor called "vi", which can be used to edit text files such as start-up scripts and configuration files.

The text editor vi assigns different meanings to buttons or keystrokes depending on the active editing mode, which can be either insert mode or command mode.

In insert mode, text typed is inserted into the document. Pressing the escape key while in insert mode will switch the editor to command mode, where letters correspond to cursor movements or editing commands.

Keys pressed while in command mode are not inserted into the text. Commands that consist of several characters, for instance ":wq", are followed by pressing enter.

*Table 7 Brief overview of vi commands*

| vi Commands | Description |
|---|---|
| i | Switch to insert mode, which is used to type text in the file. (Return to command mode by pressing the escape key.) |
| k or <arrow up> | Move cursor up |
| j or <arrow down> | Move cursor down |
| h or <arrow left> | Move cursor left |
| l or <arrow right> | Move cursor right |
| /pattern | Search forward for a pattern |
| n | Repeat previous search |
| :w | Save file |
| :w filename | Save to new filename |
| :wq | Save file and exit vi |
| ZZ | Save file if there are changes and exit vi |
| :q! | Exit vi without saving |

Type "vi" followed by the name of the file to edit. If the name of the file does not exist, a new empty file is created with that name.

# 7.5 Managing Linux Processes

A Linux process is an executing software module, for example a web server, tagd, or Reader application software.

## 7.5.1 Active Processes

There are several active processes running in a Linux system. The command "ps" displays the status for active processes.

To show process status type "ps" as follows:

```
tag:~$ ps
  PID  Uid     VmSize Stat Command
    1 root        468 S    init
    2 root            SWN  [ksoftirqd/0]
    3 root            SW   [watchdog/0]
    4 root            SW<  [events/0]
    5 root            SW<  [khelper]
    6 root            SW<  [kthread]
   16 root            SW<  [kblockd/0]
   19 root            SW<  [khubd]
   50 root            SW   [pdflush]
   51 root            SW   [pdflush]
   53 root            SW<  [aio/0]
   52 root            SW   [kswapd0]
   54 root            SW   [cifsoplockd]
   55 root            SW   [cifsdnotifyd]
  664 root            SW   [mtdblockd]
  679 root            SW<  [rpciod/0]
  723 root        772 S    /usr/sbin/sshd
  737 root        468 S    /tag/bin/tagd
```

The process list varies depending on the system; the listing above is just an example.

The leftmost column displays the Process ID (PID), which is used when referring to a process, for instance when terminating it. The rightmost column displays the command used to start the process. For example, the last row shows the tagd process with PID 737.

## 7.5.2 Terminating a Process

The command "kill" is used to send a signal to a process and terminate it.

To terminate the tagd process (PID 737) from the listing in section 7.5.1, type "kill" followed by the PID, as follows:

```
tag:~$ kill 737
```

**Note:** Terminating a process can affect other processes that are dependant on the terminated process. Only terminate a process when aware of the consequences.

## 7.5.3 Starting Processes in the Background

Processes can be run in the background, meaning that the command promt is returned after the process has been started, without having to wait for the process to finish.

The ampersand sign (&) is typed at the end of a command line to start the process in the background. To start the tagd process in the background type as follows:

```
tag:~$ /tag/bin/tagd &
tag:~$
```

### 7.5.4 Rebooting the Reader

The Reader can be hard reset by removing and reconnecting the power or soft rebooted using the command "reboot".

To restart the Reader under software control type "reboot" as follows:

```
tag:~$ reboot
Starting pid 746, console /dev/ttyS0:
'/etc/init.d/rc.reboot'
The system is going down NOW !!
Sending SIGTERM to all processes.
Requesting system reboot.
Restarting system.
```

Rebooting a Reader from a remote terminal (for instance a SSH client) will close the current network connection and temporary network settings of the Readers are lost, which can result in an unreachable Reader.

Rebooting the Reader using the service interface is recommended, so that the boot-up process can be monitored and adjusted easily.

Reboot the Reader after updating the software in order to apply changes to the configuration.

## 7.6 Configuration Scripts

This subsection describes configuration scripts that control the behaviour and the initial start-up of the Reader. The configuration scripts are plain text files, which can be edited using an ordinary text editor as described in section 7.4.7 Editing Text Files.

Take caution when editing files; only edit configuration scripts when aware of the consequences.

*Table 8 Initialisation scripts overview*

| Path | Description |
|---|---|
| /etc/inittab | Configuration file used by the init process that determines which processes are started during boot. |
| /etc/init.d/rc.sysinit | The main system initialisation script that is started by the init process. It is strongly recommended not to edit this file. |
| /etc/init.d/rc.reboot | Before the system reboots this script is run. |
| /etc/init.d/eth0.sh | Performs network configuration for device "eth0" |
| /etc/init.d/hostname.sh | Sets Reader hostname to the name stored in "/etc/hostname" |
| /tag/etc/hw_init.sh | Initialises Reader specific hardware. It is strongly recommended not to edit this file. |
| /tag/etc/sw_init.sh | Initialises system software, such as loading kernel modules and start tagd. It is strongly recommended not to edit this file. |
| /tag/etc/defaultapp | Defines which Reader application software is started when the Reader boots. |
| /etc/hosts | Host table, see section 7.7.3 Host Table |
| /etc/resolv.conf | Resolver configuration file, see section 7.7.4 |

### 7.6.1 Init Script

The init process is the last step in the boot procedure. System processes, as defined in the "/etc/inittab" file, are started during the init process.

The first process that init starts is a script that initialises the Reader system, including setting up TagMaster specific hardware, starting TagMaster specific software, configuring network interfaces, starting the web server, and setting the system time.

### 7.6.2 Reboot Script

Upon shutdown, init controls the sequence and processes for shutdown. The init process is never shut down.

### 7.6.3 Defaultapp Script

The Reader can be configured so that a specified piece of application software automatically starts when the Reader boots.

A piece of custom-made application software, in the form of an executable file, is downloaded to the Reader and for example placed in the "/tag/bin/" directory.

For information about how to transfer Reader application software, see the Software Development Kit Manual [14].

The script "/tag/etc/defaultapp" contains the full path to the Reader application software that will auto-start when the Reader boots. The script is modified to specify the default start-up application software.

Instead of using "vi", the defaultapp script that only contains one row of text, can easily be inspected and modified using the "cat" and "echo" commands.

Use the command "cat" to verify what application software is the current default start-up application, as follows:

```
tag:~$ cat /tag/etc/defaultapp
/tag/bin/wiseman &
```

In the example above, WiseMan is the default application. Use the command "echo" to modify the defaultapp script and change the default application, as follows:

```
tag:~$ echo "/tag/bin/myapp &" > /tag/etc/defaultapp
```

In the example above, "myapp" is made the default start-up application. Replace the path "/tag/bin/myapp" with the full path to any Reader application software.

Note that the ampersand sign (&) is included at the end of a command line in the defaultapp script, in order to start the application software as a background process (see section 7.5.3 Starting Processes in the Background).

## 7.7 Network Configuration

This subsection describes network configuration concepts and briefly how to configure a Reader with a TCP/IP network. However, it does not give a complete description of a TCP/IP network configuration.

Certain information is needed before configuring a Reader to run TCP/IP. At the very least, every Reader must have a unique IP address and a subnet mask.

Decide on the items described in the table below before configuring the Reader network settings.

*Table 9 Network terminology*

| Item | Description |
|---|---|
| IP address | Every network interface on a TCP/IP network must have a unique IP address. If the device is part of the Internet, its IP address must be unique within the entire Internet. If a device's TCP/IP communications are limited to a local network, its IP address only needs to be unique locally. |
| Hostname | The unique name by which a Reader is known on a network. |
| Subnet | A range of logical addresses within an IP address space |
| Subnet mask | To communicate properly, each device (Reader or host) on a network must use the same subnet mask. The subnet mask is by default set to "255.255.255.0". |
| Broadcast address | An IP address that allows information to be sent to all devices on a given subnet rather than a specific device. To avoid broadcast problems, the broadcast address of every device on a network must be the same. |
| Default gateway address | If the Reader communicates with devices that are not on its local network, a default gateway address is needed. |
| Name server addresses | To resolve hostnames into IP addresses, each Reader needs to know the addresses of one or several domain name servers. |
| Domain name | Readers using the domain name service must know their domain name. |
| Routing | Selecting paths in a computer network along which to send data. |

Regardless of whether or not the network is connected to the Internet, certain basic information must be provided to configure the network interface of the Reader. At a minimum, the network interface needs an IP address and a subnet mask.

Adding the desired configuration commands to the "eth0" initialisation script file "/etc/init.d/eth0.sh", allows automatic network configuration when the Reader starts. Temporary network settings not saved in the "eth0" initialisation script are lost when the Reader is rebooted.

Testing the configuration settings prior to modifying the script, by typing the desired commands at the line, is recommended.

Configuring the Reader for a TCP/IP network as follows:

1. Log on to the Reader as described in section 7.2 Logging on.
2. Use the ifconfig command to configure the network interface of the Reader as described in section 7.7.1 The ifconfig Command.
3. Add the address of a default gateway to the Reader's routing table as described in section 7.7.2 Routing.

4. Test the network configuration and then append the network configuration commands to "eth0" initialisation script file (/etc/init.d/eth0.sh). Edit the script file as described in section 7.4.7 Editing Text Files.
5. Reboot the Reader and verify that the saved network configuration settings are in effect.

### 7.7.1 The ifconfig Command

The ifconfig command configures the Ethernet interface. The ifconfig command will set the IP address, the subnet mask, and the broadcast address for the Ethernet interface.

Configure the Ethernet interface on a network attached Reader by typing the following command and arguments at the command line:

```
tag:~$ ifconfig eth0 192.168.0.2 netmask 255.255.255.0
```

Several different arguments can be used with the ifconfig command. A few important arguments that provide the basic information required by the network interface are described in the table below.

*Table 10 Arguments to the ifconfig command*

| Argument | Description |
|---|---|
| <interface> | The name of the network interface to configure for TCP/IP. For Readers the Ethernet interface is "eth0". |
| <address> | The IP address assigned to this interface. Enter the IP address in dotted decimal form.<br><br>The example above uses the numeric IP address "192.168.0.2" as the address value, which is also the default Reader IP address. |
| netmask <mask> | The subnet mask for this interface. Ignore this argument only if the default mask derived from the traditional address class structure is used.<br><br>The subnet mask chosen in the example above is "255.255.255.0". |

The Reader's network interface is called "eth0". That name is the first argument on every ifconfig command line. The values for the address and subnet mask are usually provided by the network administrator.

### 7.7.2 Routing

The route command manipulates the Linux kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface.

If other networks are not required to be reachable, this section can be ignored, for instance if all Readers are installed on a LAN.

The route command without parameters displays the current routing table. To add a default gateway type the following at the command line:

```
tag:~$ route add default gw 192.168.0.1
```

Note that the interface must be setup with ifconfig before a route can be added or altered.

### 7.7.3 Host Table

The host table is a simple text file that associates IP addresses with hostnames. The table is in the file "/etc/hosts". Each table entry in the host file contains an IP address separated by whitespace from a list of hostnames associated with that address. Comments begin with "#".

The host table on a Reader can for example contain the following entries:

```
#
# Table of IP addresses and hostnames
#
127.0.0.1       localhost
192.168.0.1     gateway
192.168.0.2     frontdoor
192.168.0.3     backdoor
```

Although the host table system has been superseded by DNS, the host table system is still usable for local network installations. Instead of using IP addresses the more human readable names can be used.

Pinging (see section 7.7.5) the Reader with IP address 192.168.0.3 is the same as sending a ping to the Reader "backdoor". The IP address can be replaced with the host name, as long as it is specified in the host file.

### 7.7.4 Domain Name System

If Domain Name System (DNS) support is not required, skip this section.

Very briefly, DNS can be described as a service that translates human readable domain names (hostnames) to IP addresses. For instance the domain name "tagmaster.com" is translated to the IP address 195.149.123.129.

The resolver is a set of routines that provide access to the DNS. The resolver is configured in the file "/etc/resolv.conf".

The resolver-only configuration example below is very simple. It is more or less a list of name servers to be queried:

```
#
# Domain name resolver configuration file
#
domain readers.com
# try primary
nameserver 192.168.0.11
# next try secondary
nameserver 192.168.0.12
```

The configuration tells the resolver to pass all queries to 192.168.0.11; if that fails, try 192.168.0.12.

Replace the IP addresses in the example to the IP addresses of the name servers to use.

### 7.7.5 Ping

Ping is a computer network tool used to test whether a particular host is reachable across an IP network. Ping works by sending ICMP "echo request" packets to the target host and listening for ICMP "echo response" replies.

Ping can be used to test the current network configuration. A pre-requisite is that a host that replies to echo requests is reachable; otherwise there will be no response to a ping.

When logged on to Reader, ping host 193.15.235.99, as follows (replace "193.15.235.99" with an IP address to a host that able to reply to pings):

```
tag:~$ ping 193.15.235.99
PING 193.15.235.99 (193.15.235.99): 56 data bytes
64 bytes from 193.15.235.99: icmp_seq=0 ttl=64 time=0.5 ms
64 bytes from 193.15.235.99: icmp_seq=1 ttl=64 time=0.4 ms
```

The host IP address can also be a domain name, in which case the DNS is must be configured and working properly.

By default, Readers reply to echo requests. It is therefore possible to ping a Reader from a host, as long as the host can reach the addressed Reader.

From a Windows command shell, ping a Reader with a specified IP address, as follows (replace "193.15.235.111" with an IP address that is reachable from the Windows host):

```
C:\>ping 193.15.235.111

Pinging 193.15.235.111 with 32 bytes of data:

Reply from 193.15.235.111: bytes=32 time < 1 ms TTL=64
Reply from 193.15.235.111: bytes=32 time < 1 ms TTL=64
Reply from 193.15.235.111: bytes=32 time < 1 ms TTL=64
```

If a Reader is unable to ping a host or vice versa, check that the network configuration is correct, IP addresses and net mask in particular.

### 7.7.6 Network Statistics

The command "netstat" is a network statistics tool that displays ingoing and outgoing network connections. To display all active TCP connectsions and the TCP and UDP ports on which the Reader is listening, type the following at the command line:

```
tag:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q  Local Address   Foreign Address State
tcp        0      0  *:tagp          *:*             LISTEN
tcp        0      0  *:http          *:*             LISTEN
tcp        0      0  *:ssh           *:*             LISTEN
tcp        0      0  localhost:tagp  localhost:3847  ESTABLISHED
tcp        0      0  localhost:3847  localhost:tagp  ESTABLISHED
udp        0      0  *:1023          *:*
```

From the output above it is possible to say that the tagd server that implements TAGP, as well as the web server (HTTP), and the SSH server, are all listening. The supported protocols (or services) are specified in "/etc/services".

At the time for this output, one of the standard Reader applications was up and running, which is shown as an established connection to the localhost talking TAGP.

## 7.8 Mounting Storage Devices

Mounting is the action of attaching a storage device to an existing directory in the file system, a so called mounting point.

This section describes how to mount the supported storage devices, such as USB mass storage devices and micro SD memory cards.

The file systems are defined in the file "/etc/fstab" and the mount points are created in the "/mnt/" directory.

Additional storage devices are automatically mounted when Reader boots if the corresponding mount command is added to the start-up scripts. The mounted devices should be un-mounted before the Reader is rebooted, by adding the corresponding un-mount command to the reboot script.

After mounting the file system of a storage device, the files of the device are available in the specified directory in the Reader file system.

### 7.8.1 Mounting USB Devices

The Linux operating system of the Reader is prepared with a mount point and software drivers for USB mass storage devices. The mount point for USB devices is "/mnt/usb-storage/".

Mount a USB mass storage device as follows:

1. Log-on to the Reader as described in section 7.2 Logging on, using terminal emulation software.
2. Connect the mass storage device to the USB host on the controller board. Information about the device will be displayed in the terminal emulator.
3. Type "mount /mnt/usb-storage/" followed by enter to mount the USB mass storage device.

```
tag:~$ mount /mnt/usb-storage/
```

4. Un-mount the mass storage device as follows:

```
tag:~$ umount /mnt/usb-storage/
```

Note that if there are errors when trying to un-mount a device, make sure that the current working directory is not in the file system of that device, or a file residing in that file system is open.

### 7.8.2 Mounting Micro SD Devices

The Linux operating system of the Reader is prepared with a mount point and software drivers for Micro SD Memory Cards. The mount point for Micro SD devices is "/mnt/sd-card/".

Note that the memory card must be inserted into memory card slot before Reader is powered on, otherwise the Reader will not be able to detect the memory card.

Mount memory card file system as follows:

1. Log-on to the Reader as described in section 7.2 Logging on, using terminal emulation software.
2. Type "mount /mnt/sd-card/" followed by enter to mount the memory card file system.

```
tag:~$ mount /mnt/sd-card/
```

3. Un-mount the memory card file system as follows:

```
tag:~$ umount /mnt/sd-card/
```

Note that if there are errors when trying to un-mount a device, make sure that the current working directory is not in the file system of that device, or a file residing in that file system is open.

## 7.9 SQLite Database Management

The database data are files that reside in the Reader's file system. Standard TagMaster Reader applications store the database files in "/tag/var", but the directory depends on the application.

The SQLite distribution comes with a standalone command-line access program called sqlite3, which can be used to administer an SQLite database.

The database tool is started with a path to the database file to manage, as shown below (replace "/tag/var/wiseman.db " with the path to the database file that is to be managed):

```
tag:~$ sqlite3 /tag/var/wiseman.db
```

Since the database data is stored in a regular file, backing up the database from a Reader is straightforward, just make a copy of that file.

If several Readers use the same ID-tag database, the database file can be prepared on one Reader and then copied to all the other Readers. The SQLite database

management tool is also available for hosts, so preparing a database file on a host and then downloading it to one or several Readers is also an option.

More information about managing sqlite3 is available at <http://www.sqlite.org>.

## 7.10 System Information

System information is possible to retrieve from a Reader, such as kernel version, memory usage, and tag filter status. The proc file system is a source for most information.

### 7.10.1 Web interface

The web server of the Reader provides a web interface for receiving system information such as system time, system up time, installed application software, hardware data, and memory status.

Connect to the web server of the Reader using a standard web browser installed on a PC that is connected to the same network as the Reader, as follows:

1. Configure the network settings of the PC so that the Reader is reachable over the network.
2. Open a web browser on the PC and point it to the IP address of the Reader. The default IP address of the Reader is "192.168.0.2".
3. When prompted, Log on using "root" as the user name and the corresponding password. The default password is "qwerty".
4. Click "System Information" on the initial web page to receive a status report.

If the Reader runs application software that can be configured through the web interface, an option to configure the application software will also be displayed on the initial web page. For information about configuring the Reader application software through the web interface, see WiseMan Software Manual [8] and WatchMan Software Manual [9].

### 7.10.2 Linux Kernel Version

Information about the Linux kernel version is outputted when the Reader boots. Just after the Linux kernel has been de-compressed and before it is started, U-Boot prints information about the kernel.

When logged on to a Reader it is also possible to use the command "uname" to find the kernel version as follows:

```
tag:~$ uname –a
Linux tag 2.6.16.16-at91-tm #14 Thu Nov 2 12:35:47 CET
2006 armv4tl unknown
```

The returned information contains Linux kernel version name and when it was built.

### 7.10.3 tagmod Information

The tagmod kernel module provides information via the proc file system, which is useful information when contacting the TagMaster technical support. The information contains for instance the Reader controller board serial number.

When logged on to a Reader display the tagmod information as follows:

```
tag:~$ cat /proc/tagmod
Board serial no  : 063500042
Board revision   : 02
Product type no  : 07
Module version   : v1.1.0
Build date       : Nov 29 2006 17:00:41
IRQ counter      : 45
Unknown IRQs     : 0
Tag write IRQs   : 0
FPGA IRQ words   : 4556
Too big          : 0
Too small        : 4
Queue wraps      : 0
Queue reads      : 0
Queue r bytes    : 0
Queue writes     : 0
Queue w bytes    : 0
Mark CRC errors  : 41
Data CRC errors  : 0
Frequency hops   : 0
Debug output     : OFF
Mark CRC discard : ON
Data CRC discard : ON
Read beep        : OFF
FHSS             : OFF
```

This information is also available via the web interface.

### 7.10.4 ID-tag Filter Information

When logged on to a Reader it is also possible to display information about the ID-tag filter. The information contains the number of read ID-tags, how many ID-tags that have been filtered and so forth. ID-tags residing in the filter, if any, are also displayed.

Display the ID-tag filter information as follows:

```
tag:~$ cat /proc/tagfilter
Timeout   : 0 ms (0 jiffies)
Mode      : off
Depth     : 50 tags
In filter : 0 tags
Processed : 0 tags
Filtered  : 0 tags (0%)
Time now  : 4294962968 jiffies
```

This information is also available via the web interface.

### 7.10.5 Memory Usage

The current memory usage, both RAM and Flash, can be retrieved when logged on to a Reader. To get information about available RAM memory type "free", as follows:

```
tag:~$ free
             total       used       free     shared    buffers
  Mem:        29968       6908      23060          0          0
 Swap:            0          0          0
Total:        29968       6908      23060
```

The returned information shows the total amount of available memory, used memory and free memory. A Reader running the default services and standard Reader application software will have most of the available memory free.

Swap is always set to 0, because Readers do not use swap memory.

Using the command "df", it is also possible to get information about the file system memory usage, as shown below:

```
tag:~$ df
Filesystem     1k-blocks  Used  Available Use%  Mounted on
/dev/mtdblock0     14336  3960      10376  28%  /
```

# 8 Software Updates

All Reader software including operating system, root file system, and boot loader can be updated through the service interface.

Files residing in the file system, such as FPGA configuration data, drivers, and applications can additionally be updated through the Ethernet interface.

## 8.1 Updating Files in the File System

The Reader file system contains application software, device drivers, and configuration files. These software components are single files, which can be updated separately.

The file system can be accessed through the Ethernet interface using Secure File Transfer Protocol (SFTP) or Secure Copy Protocol (SCP). The service interface using the XMODEM protocol can also be used to transfer files. The network interface is recommended when updating files.

Before updating a file, kill all processes related to that file as described in section 7.5.2 Terminating a Process.

After updating a file, ensure that file access permissions are set correctly. Changing file access permissions are described in section 7.4.6 Change Mode.

**Note:** If transferring the file fails, verify that no processes are associated with the file being replaced and that the access permissions allow overwriting the file. If it is impossible to connect to a Reader from a PC, verify that the Reader is powered and that the Reader is reachable via the network.

### 8.1.1 Updating Files using Ethernet and SFTP

When updating files through the Ethernet interface, SFTP can be used. Use an SFTP client on the PC, which communicates securely over the Ethernet interface with the Reader.

Update files in the Reader file system using SFTP as follows:

1. Configure the SFTP-client with the following communication settings:
    - Host is set to the IP address of the Reader (default IP address is 192.168.0.2)
    - Port 22
    - Servertype is set to SFTP using SSH
    - Username "root" and the current password
2. Open a connection from the SFTP-client to the Reader. A file manager is displayed showing both the local file system of the PC and the remote file system of the Reader.
3. Locate the update file in the local file system and transfer it to the designated location in Reader file system, replacing the old file.

### 8.1.2 Updating Files using Ethernet and SCP

SCP is an alternative to SFTP, which is usually a command line tool for copying files to and from a PC and a Reader.

The SCP tool is usually installed on Unix and Linux systems by default. Windows systems usually require that the SCP tool is installed separately. See 12Appendix A for suggested application software.

Update a file in the Reader file system using SCP as follows (terminal output from a typical Linux system):

```
user@debian:~$ scp sourcefile root@192.168.0.2:/tag/
```

The example above copies the file "sourcefile" to the directory "/tag/" on Reader with IP address 192.168.0.2. The user will be prompted for password before the file copy process is started.

SCP may seem more rudimentary than SFTP but it is useful when creating scripted batch jobs, for instance automatically updating several files residing on different Readers.

### 8.1.3 Updating Files using Service Interface and XMODEM

Updating files using the service interface requires direct access to the service interface and terminal emulation software that supports file transfer using the XMODEM protocol.

Update files in the Reader file system using XMODEM as follows:

1. Log-on to the Reader as described in section 7.2.1 Logging on Using Service Interface.
2. Type "rx" followed by the name of the file to update, including the complete path. The Reader is ready to receive the file when the service interface periodically outputs the character "C".
3. Click send file using the XMODEM protocol in the terminal emulation software and select file to send. How to actually do this depends on the terminal emulation software.
4. After file download is completed the command line is returned on the service interface. Changing the file mode might be necessary, as described in section 7.4.6 Change Mode.

## 8.2 Upgrade Flash Memory using U-Boot

This section describes how to upgrade the U-Boot image residing in Flash using U-Boot. The same instruction applies when upgrading any image in the Flash, just change address, memory address range and file name accordingly.

The Flash memory is divided into five image areas as seen in the table below.

*Table 11 Flash memory map*

| Memory address range | Description |
|---|---|
| 0x10000000 – 0x1001ffff | Initial boot loader, typically a file named "flash_1_2_0_boot.bin" |
| 0x10020000 – 0x1003ffff | Gzipped boot loader, typically a file named "flash_1_2_0_u-boot.bin.gz" |
| 0x10040000 – 0x1005ffff | U-Boot environment variables, which is only used by U-Boot |
| 0x10060000 – 0x101fffff | Linux kernel, typically a file named "flash_1_2_0_linuxkernel-2.6.16.16" |
| 0x10200000 – 0x10ffffff | File system, typically a file named "flash_1_2_0_fs.img" |

If U-Boot is not available due to reasons such as corrupted Flash, erased flash, and so forth, see section 8.3 Restore U-Boot or Initial Boot Loader.

Upgrade Flash memory as follows:

1. Enter U-Boot user interface by pressing any key and aborting auto-boot.
2. Get binary as described in section 8.2.1 Download File.
3. Unprotect Flash sectors as described in section 8.2.2 Prepare Flash.
4. Copy the downloaded data to the Flash as described in section 8.2.3 Copy to Flash and Verify

### 8.2.1 Download File

Get binary, either using kermit serial protocol or tftp. Note that tftp download requires a tftp-server and that network has been configured. The TagMaster SDK provides a pre-configured tftp-server.

Download boot loader binary to RAM, at address 0x21000000 using kermit protocol. Type "loadb 21000000" and then select send file using kermit in the terminal emulator software, as follows:

```
U-Boot> loadb 21000000
## Ready for binary (kermit) download to 0x21000000 at
115200 bps...
## Total Size      = 0x0000bed7 = 48855 Bytes
## Start Addr      = 0x21000000
```

Or download boot loader binary to RAM, at address 0x21000000 using tftp, as follows:

```
U-Boot> tftp 21000000 u-boot.bin.gz
TFTP from server 193.15.235.99; our IP address is
193.15.235.202
Filename 'u-boot.bin.gz'.
Load address: 0x21000000
Loading: ##########
done
Bytes transferred = 48855 (bed7 hex)
```

Note the size written as a hexadecimal value "bed7", which is the size of the downloaded file. This value is required when writing the file content to the Flash memory. The value will of course depend on the file that is downloaded and will most likely differ for all files.

### 8.2.2 Prepare Flash

Unprotect Flash sectors and erase them:

```
U-Boot> protect off 10020000 1003ffff
.
Un-Protected 1 sectors
U-Boot> erase 10020000 1003ffff

. done
Erased 1 sectors
```

### 8.2.3 Copy to Flash and Verify

Copy the downloaded data to the Flash and verify that the copying was successful. Note the size written as a hexadecimal value "bed7", which is the size of the downloaded file. It was specified after the download was completed.

```
U-Boot> cp.b 21000000 10020000 bed7
Copy to Flash... done
U-Boot> cmp.b 21000000 10020000 bed7
Total of 48855 bytes were the same
```

## 8.3 Restore U-Boot or Initial Boot Loader

If the initial boot loader or U-Boot does not start, one or both boot loaders must be restored.

If this occurs, contact TagMaster technical support for more information.

# 9 Contact

For any further inquiries, please contact TagMaster AB.

## 9.1 Technical Support

Phone: + 46 8 632 19 50
Fax: +46 8 750 53 62
E-mail: support@tagmaster.com

## 9.2 Office

TagMaster AB
Kronborgsgränd 1
S-164 87  KISTA, Sweden

Phone: +46 8 632 19 50
Fax: +46 8 750 53 62
E-mail: sales@tagmaster.com
Web: www.tagmaster.com

# 10 Glossary

| | |
|---|---|
| **ARM9** | A 32-bit RISC processor architecture that is widely used in a number of embedded designs. |
| **BusyBox** | Combines tiny versions of many common UNIX utilities into a single small executable |
| **CGI** | Common Gateway Interface<br>A standard protocol for interfacing external application software with a web server. |
| **FHSS** | Frequency-Hopping Spread Spectrum<br>A spread-spectrum method of transmitting radio signals by rapidly switching a carrier among many frequency channels, using a pseudorandom sequence known to both transmitter and receiver. |
| **Flash memory** | A form of non-volatile computer memory that can be electrically erased and reprogrammed. |
| **FPGA** | Field Programmable Gate Array |
| **GPL** | GNU General Public License<br>A widely-used free software license |
| **ICMP** | Internet Control Message Protocol<br>Core Internet protocol used by networked computers to send error messages |
| **ID-tag** | ID-carrier in the TagMaster system, which is readable and writable via the radio interface. |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **JFFS2** | Journaling Flash File System version 2 |
| **multi-drop network** | A network in which several Readers are all connected to the same host. |
| **Open32** | TagMaster ID-tag format |
| **Reader application software** | The application software installed in the Reader, defining the behaviour of the Reader. |
| **Reader system software** | The fundamental software and operating system installed in the Reader. |
| **reading lobe** | The space in front of the Reader in which ID-tags are being read. |

| | |
|---|---|
| **SFTP** | Secure File Transfer Procol |
| **SSH** | Secure Shell |
| **tagd** | TagMaster daemon process |
| **taglib** | TagMaster software library |
| **tagmod** | TagMaster kernel module |
| **TAGP** | TagMaster network communication protocol |
| **XMODEM** | A file transfer protocol |
| **ConfiTalk** | TagMaster standard protocol for serial communication. |

# 11 References

The following documents can be downloaded from TagMaster's web site <http://www.tagmaster.com>.

**1** *LR-6 Reader Data Sheet*
Doc no. 06-218

**2** *LR-series Reader Installation Manual*
Doc no. 06-136

**3** *S1251 ScriptTag Data Sheet*
Doc no. 04-182

**4** *S1255 MarkTag Data Sheet*
Doc no. 04-219

**5** *S1450 ScriptTag HD Data Sheet*
Doc no. 04-219

**6** *S1455 MarkTag HD Data Sheet*
Doc no. 04-187

**7** *S1456 MarkTag HDS Data Sheet*
Doc no. 04-189

**8** *WiseMan Software Manual*
Doc no. 06-138

**9** *WatchMan Software Manual*
Doc no. 06-239

**10** *PassMan Software Manual*
Doc no. 06-240

**11** *TAGP Protocol Specification v1.1*
Doc no. 05-172

**12** *taglib Software Library Specification v1.2*
Doc no. 05-249

**13** *ConfiTalk Reference Manual*
*Doc no. 510213*

**14** *Software Development Kit Manual*
Doc no. 06-195

© TagMaster AB 

# 12 Online Resources

| | |
|---|---|
| http://www.gnu.org | The GNU Project |
| http://www.kernel.org | Primary site for the Linux kernel source |
| http://u-boot.sourceforge.net | The Das U-Boot Free Software project |
| http://www.sqlite.org | SQLite home page |
| http://sourceforge.jp/projects/ttssh2/ | UTF-8 TeraTerm Pro with TTSSH2 download site |
| http://www.busybox.net | BusyBox: The Swiss Army Knife of Embedded Linux |

# Appendix A Useful Software Tools

The following tables list software tools, useful when working with a network-connected Reader.

A terminal emulator is a program that emulates a video terminal within some other display architecture. Though typically synonymous with a command line shell or text terminal, the term terminal covers all remote terminals, including graphical interfaces.

*Table 12 Terminal emulator software*

| Application | Description |
|---|---|
| HyperTerminal | Bundled with multiple versions of Microsoft Windows (not Windows Vista) |
| Tera Term | Open source free software terminal emulator for Windows <http://sourceforge.jp/projects/ttssh2> |

Secure Shell or SSH is a set of standards and an associated network protocol that allows establishing a secure channel between a local and a remote computer. It uses public-key cryptography to authenticate the remote computer and (optionally) to allow the remote computer to authenticate the user.

*Table 13 SSH Clients*

| Application | Description |
|---|---|
| Secure Shell (SSH) | Allows establishing a secure channel between a local and a remote computer <ftp://ftp.ssh.com> |
| Putty | A free implementation of SSH and SCP for Windows and Unix platforms <http://www.chiark.greenend.org.uk/~sgtatham/putty> |

SSH File Transfer Protocol (SFTP) is a network protocol that provides file transfer and manipulation functionality over any reliable data stream. It is typically used with the SSH-2 protocol to provide secure file transfer.

*Table 14 SFTP Client*

| Application | Description |
|---|---|
| FileZilla | Open source free FTP client for Windows. It supports both FTP and SFTP |

© TagMaster AB 